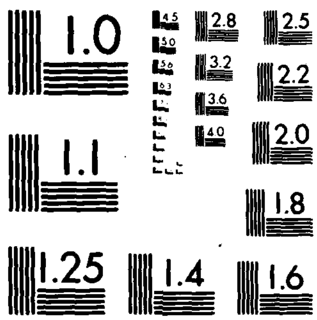
AD-A092 750 STANFORD UNIV CA SYSTEMS OPTIMIZATION LAB F/G 12/1
TIME-STAGED METHODS IN LINEAR PROGRAMMING, COMMENTS AND EARLY H--ETC(U)
JUN 80 G B DANTZIG N00014-75-C-0267
UNCLASSIFIED SOL-80-18 NL

END
DATE
FILMED
1-81
DTIC



MICROCOPY RESOLUTION TEST CHART

(2)

Systems
Optimization
Laboratory

LEVEL II

AD A092750

DDC FILE COPY

DTIC
ELECTE
DEC 10 1980
S D
E

DISTRIBUTION STATEMENT A
Approved for public release;
Distribution Unlimited

Department of Operations Research
Stanford University
Stanford, CA 94305

80 12 01 29

LEVEL II

(2)

SYSTEMS OPTIMIZATION LABORATORY
DEPARTMENT OF OPERATIONS RESEARCH
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305

TIME-STAGED METHODS IN LINEAR PROGRAMMING
Comments and Early History

by

George B. Dantzig

TECHNICAL REPORT SOL 80-18

June 1980

Research and reproduction of this report were supported by the Department of Energy Contract DE-AC03-76SF00326, PA# DE-AT03-76ER72018, Office of Naval Research Contract N00014-75-C-0267, National Science Foundation Grants MCS76-81259, MCS7926009 and ENG77-06761. This paper was presented at "Workshop on Large-Scale Linear Programming", June 2-6, 1980, International Institute for Applied Systems Analysis, Austria.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: SOL-80-18

2. GOVT ACCESSION NO.: AD-A092750

3. RECIPIENT'S CATALOG NUMBER:

4. TITLE (and Subtitle): 6 TIME-STAGED METHODS IN LINEAR PROGRAMMING, Comments and Early History.

5. TYPE OF REPORT & PERIOD COVERED: 9 Technical Report,

6. PERFORMING ORG. REPORT NUMBER:

7. AUTHOR(s): 10 George B. Dantzig

8. CONTRACT OR GRANT NUMBER(s): 15 N00014-75-C-0267, DE-AC03-76-SF-00326

9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Department of Operations Research - SOL
Stanford University
Stanford, CA 94305
408 765

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR-047-143

11. CONTROLLING OFFICE NAME AND ADDRESS:
Office of Naval Research - Dept. of the Navy
800 N. Quincy Street
Arlington, VA 22217

12. REPORT DATE: 11 Jun[redacted]80

13. NUMBER OF PAGES: 13

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 12 18

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

LINEAR PROGRAMMING HISTORY
DYNAMIC L.P.
TIME STAGED
LARGE SCALE
STRUCTURED

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

see attached

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

408765 Du

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SOL 80-18: Time-Staged Methods in Linear Programming: Comments and Early History, by George B. Dantzig

This Workshop on Large-scale Mathematical Programs reflects the active research taking place in many parts of the world along a very broad front-- namely:

(1) on the theory of solution,
(2) on software development,
(3) on experiments on representative problems,
(4) on application to real problems,
(5) on matrix input generators,
(6) on matrix analyzers,
(7) on output report generators,
(8) on alternative methods of formulation.

This paper is a historical review of the author's interest in one important facet of this field - namely the solution of time-staged programs. Indeed it was dynamic L.P. that initiated the linear programming field back in 1947. Over the years, many good ideas have been proposed, some that still merit serious consideration. This Workshop may provide the answer to the question whether or not we have begun at last to achieve the efficiency of solution necessary for successful application.

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

TIME-STAGED METHODS IN LINEAR PROGRAMMING
COMMENTS AND EARLY HISTORY

George B. Dantzig

June 1980

ABSTRACT

This Workshop on Large-scale Mathematical Programs reflects the active research taking place in many parts of the world along a very broad front-- namely

(1) on the theory of solution
(2) on software development
(3) on experiments on representative problems
(4) on application to real problems
(5) on matrix input generators
(6) on matrix analyzers
(7) on output report generators
(8) on alternative methods of formulation

This paper is a historical review of the author's interest in one important facet of this field - namely the solution of time-staged programs. Indeed it was dynamic L.P. that initiated the linear programming field back in 1947. Over the years, many good ideas have been proposed, some that still merit serious consideration. This Workshop may provide the answer to the question whether or not we have begun at last to achieve the efficiency of solution necessary for successful application.

TIME-STAGED METHODS IN LINEAR PROGRAMMING
COMMENTS AND EARLY HISTORY

This paper is a more polished version of the talk which I delivered opening the International Institute for Applied Systems Analysis Workshop on Large-Scale Linear Programming at Laxenburg Austria, June 2-6, 1980. Except for a short review of large-scale methods also presented, but omitted here, my perspective is historical.

TIME-STAGED STAIRCASE SYSTEMS

The first formal papers about the new field of linear programming (that started in 1947) appeared in Econometrica July-October 1949. At the very beginning, the emphasis was on solving time-staged (dynamic) linear programs. That this is so, is clear from the following quote from [1]:

> This paper is concerned with improved techniques of program planning, particularly as they apply to the scheduling of activities over time within an organization or economy in which the activities must share in the use of limited amounts of various commodities. The contemplated use of electronic computers for rapidly computing programs and the assumptions underlying the mathematical model are discussed. The paper is concluded by an illustrative example, [Berlin Airlift, A Time-Staged Dynamic Linear Program].
>
> The Mathematical Model discussed here is a generalization of the Leontief Inter-Industry Model. It is closely related to the one found in von Neumann's paper "A Model of General Economic Equilibrium". Its chief points of difference lie in its emphasis on dynamic, rather than equilibrium or steady states. Its purpose is close control of an organization--

hence it must be quite detailed; it is designed to handle highly dynamic problems--hence greater emphasis on time lags and capital equipment; it takes into consideration the many different ways of doing things--hence it explicitly introduces alternative activities; and it recognizes that any particular choice of a dynamic program depends on the "objectives" of the "economy",--hence the selection and types of activities are made to depend on the maximization of an objective function.

In the companion paper [2], the time staged staircase model is displayed and its relationship to Leontief Input-Output model and continuous-time models is discussed:

$$
\begin{array}{lllllll}
\alpha^{(1)}x^{(1)} & \cdots & & & & = & a^{(1)} \\
-\bar{\alpha}^{(1)}x^{(1)} & +\alpha^{(2)}x^{(2)} & \cdots & & & = & a^{(2)} \\
\cdots & -\bar{\alpha}^{(2)}x^{(2)} & +\alpha^{(3)}x^{(3)} & \cdots & & = & a^{(3)} \\
\cdots & \cdots & \cdots & \cdots & \cdots & & \cdots \\
 & & & -\bar{\alpha}^{(T-1)}x^{(T-1)} & +\alpha^{(T)}x^{(T)} & = & a^{(T)} \\
\gamma^{(1)}x^{(1)} & + \cdots & & & +\gamma^{(T)}x^{(T)} & = & \max,
\end{array}
$$

where the $x^{(t)}$ are vectors of nonnegative elements.

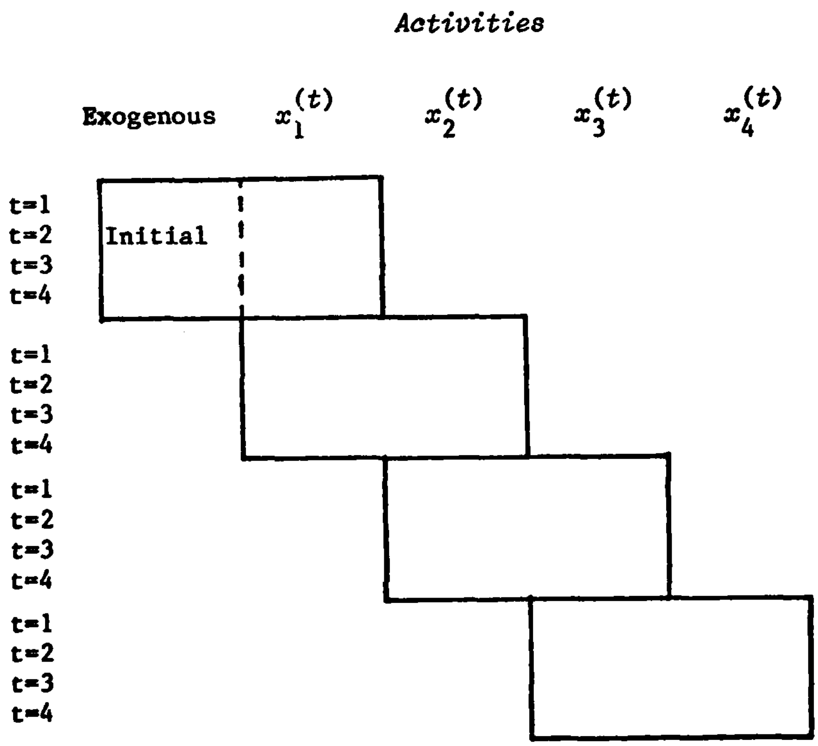
When the matrices $\alpha^{(t)}$ and $\bar{\alpha}^{(t)}$ $(t=1,2,\ldots,T)$ are square and nonsingular, a direct solution is possible that may lead, however, to negative and nonnegative activity levels (in which case no feasible solution exists).

It should be noted that the general mathematical problem reduces in the linear programming case to consideration of a system of equations of nonnegative variables whose matrix of coefficients is composed mostly of blocks of zeros except for submatrices along and just off the "diagonal". Thus any good computational technique for solving programs would probably take advantage of this fact.

Having formulated the time-staged model, it soon became clear that the techniques at hand at the time were inadequate. In a companion paper [3], first presented in 1949, appeared the following statement:

> Computing techniques are now available for solution of small linear programming problems. However, for accurate over-all Air Force planning, the size of the required model is such that conventional punched card computing equipment, or even the interim electronic computer being built for the Air Force by the National Bureau of Standards, is not sufficiently powerful to cope satisfactorily with the problem of choosing the optimum activities and activity levels over time.

In order to obtain a programming procedure which would be immediately useful with presently available computing equipment, we have been *forced* to use a determinate, and hence less general formulation of the programming problem that parallels closely the staff procedure.



We have called this a *triangular model* because in it the matrix of detached coefficients, when arragned as in the Table, and omitting the "initial" part, assumes a triangular form, with all coefficients above and to the right of the principal diagonal being zero. Thus the activities and items are so ordered that the levels of any one activity over time depend only on the levels of the activities which precede it in the hierarchy. This means that in the computation of the program we successively work down the hierarchy, at each step solving completely for the levels of each activity in each of the time periods before proceeding to the next activity (see figure above).

The triangular model technique is a powerful empirical method when there is a natural hierarchy of activities and output items. Certain energy models, for example, currently in vogue use such an approach.

BLOCK TRIANGULARITY

My paper [4], is my first on methods for solving large systems:

> With the growing awareness of the potentialities of the linear programming approach to both dynamic and static problems of industry, of the economy, and of the military, the main obstacle toward full application is the inability of current computational methods to cope with the magnitude of the technological matrices for even the simplest situations. However, in certain cases, such as the now classical Hitchcock-Koopmans transportation model, it has been possible to solve the linear inequality system in spite of size because of simple properties of the system. This suggests that considerable research be undertaken to exploit certain special matrix structures in order to facilitate ready solution of larger systems.
>
> Indeed, recent computational experience has made it clear that standard techniques such as the simplex algorithm, which have been used to solve successfully general systems involving one hundred equations (in any reasonable number of nonnegative unknowns), are too tedious and lengthy to be practical for extensions much beyond this figure. Our purpose here will be to develop short-cut computational methods for solving an important class of systems whose matrices may be generally described as "block triangular".
>
> By "block" triangular we mean that if one partitions the matrix of coefficients of the technology matrix into submatrices, the submatrices (or blocks) considered as elements form a *triangular system*,
>
> $$\begin{bmatrix} A_{11} & & & \\ A_{21} & A_{22} & & \\ \cdots & \cdots & & \\ A_{T1} & A_{T2} & \cdots\cdots & A_{TT} \end{bmatrix}$$
>
> For example, von Neumann, in considering a constantly expanding economy, developed a linear dynamic model whose matrix of coefficients may be written in the form,
>
> $$\begin{bmatrix} A & & & & \\ -B & A & & & \\ & -B & A & & \\ & & \ddots & \ddots & \\ & & & -B & A \end{bmatrix}$$
>
> where A is the submatrix of coefficients of activities initiated in period t, and B is the submatrix of *output* coefficients of these activities in the following period.

Now the main obstacle toward the full application of standard linear programming techniques to dynamic systems is the magnitude of the matrix for even the simplest situations. For example, a trivial 15-activity--7-item static model, when set up as a 12-period dynamic model, would become a 180-activity by 84-item system, which is considered a large problem for application of the standard simplex method. A fancy model involving, say, 200 activities and 100 items for a static case would become a 2000 x 1000 matrix if recast as a 10-period model. It is clear that dynamic models must be treated with special tools if any progress is to be made toward solutions of these systems.

From a computational point of view, there are a number of observed characteristics of the dynamic models which are often true for static models as well.
These are:

(1) The matrix (or its transpose) can be arranged in triangular form
(2) Most submatrices A_{ij} are either zero matrices or composed of elements, most of which are zero.
(3) A basis for the simplex method is often block triangular with its diagonal submatrices square and nonsingular (referred to as a "square block triangular" basis).
(4) For dynamic models similar type activities are likely to persist in the basis for several periods.

To illustrate, consider a dynamic version of the Leontief model in which (a) alternative activities are permitted (a simple case would be where steel can be obtained from direct production or storage); (b) inputs to an activity for production in the tth time period may occur in the same or earlier time periods. It can be shown in this model that (a) a basic solution will have exactly m activities in each time period (where m = number of time dependent equations), (b) each shift in basis will bring in a substitute activity in the same time period, and (c) optimization can be carried out as a sequence of one-period optimization problems; i.e., the optimum choice of activities (but not their amounts) can be determined for the first time period (independent of the later periods) this permits a determination for the second time period (independent of the later periods), et cetera.

When flow models are replaced with more complex models which include initial inventories, capacities, and the building of new capacities, the ideal structure of a basis (see third characteristic above) no longer holds. However, tests (carried on since 1950) on a number of cases indicate that bases, while often *not square block triangular in the sense above, could be made so by changing relatively few columns in the basis* (e.g., one or two activities in small models). This characteristic of *near-square block* triangularity of the basis, i.e., with nonsingular square submatrices down the diagonal, is, of course, computationally convenient and this paper will be concerned with ways to exploit it.

Towards the end of the above paper can be found the following:

> Finally, may I make a short plea that linear programmers pay greater attention to special methods for solving the larger matrices that are encountered in practice. The excellent work of Jacobs on the caterer problem and the work of Jacobs, Hoffman, Johnson on the production smoothing problem are examples of what may be done with certain dynamic models with a simple repetitive structure. Cooper and Charnes have employed in their work a number of short cuts that have permitted resolution of certain large scale systems. At RAND we have found efficient ways to hand compute generalized transportation problems, and Markowitz has proposed a general procedure in this area that is promising. Many models exhibit a block triangular structure and certain partitioning methods have been proposed which take advantage of this type of structure. There is need for those of you who are foresighted to do serious research in this area.
>
> At the present time (1955), it is possible to solve rapidly problems in the order of a hundred equations. The Orchard-Hays 701 Simplex Code has solved many problems of this size with as high as 1,500 unknowns and machine times of five to eight hours as a rule--all with excellent standards of accuracy. However, it is self-evident that no matter how much the general purpose codes are perfected they will be unable to cope with the next generation of problems which will be larger in size. It is also evident that the models currently being run could have been handled more effectively by the proposed special methods.
>
> There are certain characteristics common to many models which I believe should be emphasized:
> (1) Most factors in the coefficient matrix are zero.
> (2) In dynamic structures the coefficients are often the same from one time period to the next.
> (3) In dynamic solutions the activities employed often persist from one period to the next.
> (4) Transportation type submatrices are common.
> (5) Block triangular submatrices are common.
>
> Part of the research in this area should certainly be devoted to a better understanding of the potentialities of techniques other than the simplex method.

UNCERTAINTY

In a related paper [5], published in 1956, appears the following

> In the past few months there have been important developments that point to the *application of linear programming methods under uncertainty*. By way of background let us recall that there are in common use two essentially different types of scheduling applications--one designed for the short run and those

> for the long run. For the latter the effect of probabilistic or chance events is reduced to a minimum, by the usual technique of providing plenty of *fat* in the system. For example, *consumption rates, attrition rates, wear-out rates* are all planned on the high side. *Times to ship, time to travel, times to produce* are always made well above actual needs. Indeed, the entire system is put together with plenty of *slack* and *fat* with the hope that they will be the *shock absorbers* which will permit the general objectives and timing of the plan to be executed in spite of unforeseen events. In the general course of things, long-range plans are revised frequently because the stochastics elements of the problem have a nasty way of intruding. For this reason also the chief contribution, if any, of the long-range plan, is to effect an immediate decision--such as the appropiation of funds or the initiation of an important development contract.
>
> For short-run scheduling, many of the *slack* and *fat* techniques of its long-range brother are employed. The principle differences are attention to detail and the short time-horizon. As long as *capabilities* are well above *requirements* (or demands) or if the demands can be shifted in time, this approach presents no problems since it is feasible to implement the schedule in detail. However, where there are shortages, the projected plan based on such techniques may lead to actions far from optimal, whereas these new methods, where applicable, may result in considerable savings. I shall substantiate this later by reference to a problem of A. Ferguson on the routing of aircraft.
>
> With regard to the possibilities of solving large scale linear programming problems, one can sound both an optimistic and a pessimistic note.. The pessimistic note concerns the ability of the problem formulator, either amateur or professional, to develop models that are large scale. The pessimistic note also concerns the inability of the problem sovler to compute models *by general techniques* when they are large scale. If this is so, is not the great promise that the linear programming approach will solve scheduling and long range planning problems with substantial savings to the organizations adopting these methods but an illusion and a snare? Are the big problems going to be solved as they have always been solved--by a detailed system of on-the-spot somewhat natural set of priorities that resolve every possible alternative as it arises?

The status of problems involving uncertainty as far as practical solutions are concerned, has not changed much since 1956. The following, sums up the 1965 situation:

> When one considers instead, a direct attack on uncertainty via mathematical programming, it inevitably leads to the consideration of *large-scale systems*. Problems with their structure, have proven difficult of solution so far. I believe that they will be the subject of intensive investigation in the future.

DECOMPOSITION PRINCIPLE

The Decomposition Principle [6] arose in 1958 in connection with a military tactical problem which was too large to handle by conventional linear programming problem. A good summary of the approach can be found in my 1965 survey article:

> Recently the author, jointly with Philip Wolfe, developed a new procedure that is particularly applicable to angular systems and multistage systems of the staircase type This is reported in preliminary form in RAND P-1544 (Nov.10, 1958) under the title, "A Decomposition Principle for Linear Programs". The system consists of certain goods shared in common among several parts and certain goods (including facilities, raw materials) peculiar to each part. In short the system is angular in structure.
>
> Although the entire procedure is one intended to be carried out internally in an electronic computer it may also be viewed as a *decentralized decision making process*. Each independent part initially offers a possible bill of goods (a vector of the *common* outputs and supporting inputs including outside costs) to a central coordinating agency. As a set these are mutually feasible with each other and the given common resources and demands from outside the system. The coordinator works out a system of "prices" for paying for each component of the vector plus a special subsidy for each part that just balances the cost.
>
> The management of each part then offers, based on these prices, a new feasible program for his part with lower cost *without regard to whether it is feasible for the system as a whole*. The coordinator, however, combines these new offers with the set of earlier offers so as to preserve mutual feasibility and consistency with exogeneous demand and supply and to minimize cost. Using the improved over-all solution he generates a revised set of prices, subsidies, and receives new offers. The essential idea is that old offers are never forgotten by the central agency (unless using "current" prices they are unprofitable); the former are mixed with the new offers to form new prices.

In the original paper [6] appears this abstract:

> A technique is presented for the decomposition of a linear program that permits the problem to be solved by alternate solutions of linear sub-programs representing its several parts and a coordinating program that is obtained from the parts by linear transformations. The coordinating program generates at each cycle new objective forms for each part, and each part generates in turn (from its optimal basic feasible solutions) new activities (columns) for the interconnecting program. Viewed as an instance of a 'generalized programming problem' whose columns are drawn freely from given convex sets, such a problem can be studied by an appropriate generalization of the duality theorem for linear

> programming, which permits a sharp distinction to be made between those constraints that pertain only to a part of the problem and those that connect its parts. This leads to a generalization of the Simplex Algorithm, for which the decomposition procedure becomes a special case.

The reported experience with solving structured linear programs by means of the decomposition principle varies from very good to poor. In general it appears that if the decomposition between master and sub is a "natural" one, it can perform very well. Like the simplex method, there is rapid improvement for the early iterations followed by a long tail except here the tail is much longer.

COMPACT BASIS INVERSES

From 1962 onwards there has been growing interest in schemes for compactly representing the inverse of the basis for the simplex method. This effort goes under various names: compact basis triangularization, LU basis factorization. One must worry not only about the compactness but also about the stability of the solution to small changes in the original data. My 1962 paper [7] was directed to finding a compact representation of a basis for staircase systems.

> Alex Orden was the first to point out that the inverse of the basis in the simplex method serves no function except as a means for obtaining the representation of the vector entering the basis and for determining the new price vector. For this purpose one of the many forms of "substitute inverses" (such as the well known product form of the inverse) would do just as well and in fact may have certain advantages in computation.
>
> Harry Markowitz was interested in developing, for a sparse matrix, a substitute inverse with as few nonzero entries as possible. He suggested several ways to do this approximately. For example, the basis could be reduced to triangular form by successively selecting for pivot position that row and column whose product of nonzero entries (excluding the pivot) is minimum. He also pointed out that, for bases whose nonzeros appear in a band on a staircase about the diagonal, proper selection of pivots could result in a compact substitute with no more nonzeros than the original basis.
>
> We shall adopt Markowitz's suggestion. However, instead of recording the successive transformations of one basis to the next in product form, we shall show that it is efficient to generate each substitute inverse in turn from its predecessor. The substitute inverse remains compact, of fixed size. Thus "reinversions" are unnecessary (except in so far as they are needed to restore loss of accuracy due to cumulative round-off error).
>
> The procedure which we shall give can be applied to a general m x m basis without special structure. As such, it is

probably competitive with the standard product form, for it may have all of its advantages and none of its disadvantages. With certain matrix structures, moreover, it appears to be particularly attractive.

We shall focus our remarks on *staircase structures*. The reader will find no difficulty in finding an equally efficient way to compact block-angular structures.

STATUS AS OF 1967

A summary of the status of solving large-acale problems can be found in my 1967 paper [8].

From its very inception, it was envisioned that linear programming would be applied to very large, detailed models of economic and military systems. Kantorovitch's 1939 proposals, which were before the advent of the electronic computer, mentioned such possibilities. Linear programming evolved out of the U.S. Air Force interest in 1947 in finding optimal time-staged deployment plans in case of war; a problem whose mathematical structure is similar to that of finding an optimal growth pattern of a developing economy and similar to other control problems. Structurally the dynamic problems are characterized in discrete form by staircase matrices representing the inputs and outputs from one time period to the next. Treated as an ordinary linear program, the number of rows and columns grows in proportion to the number of time periods T and the computational effort grows by T^3 and possibly higher. This fact has limited the use of linear programming as a tool for planning over many time periods.

At the present 1967 stage of the computer revolution, there is growing interest on the part of practical users of linear programming models to solve larger and larger systems. Such applications imply that eventually automated systems will obtain information from counters and sensing devices, process data into the proper form for optimization and finally implement the results by control devices. There has been steady progress in this mechanization of flow to and from the computer. Hitherto, this has been one of the obstacles encountered in setting-up and solving large-scale systems. The second obstacle has been the cost and the time required to successfully solve large problems.

It is difficult to measure the potential of large-scale planning. Certain developing countries appear, according to optimal calculations on simplified models to be able to grow at the rate of 15% per year implying a doubling of their industrial base in five years. But administrators apparently ignore plans and make decisions based on political expediency which restrict growth to 2 or 3% or sometimes -2%. It is the belief of the author that the mechanization of data flow (at least in advanced countries) in the next decade will provide pathways for constructing

> large models and the effective implementation of the results of optimization. This application of mathematics to decision processes will eventually become as important as the classical applications to physics and will, in time, change the emphasis in pure mathematics.

In this paper the following unsolved problem was posed:

> It has been discovered recently that the size of the inverse representation of the basis in the simplex method could have an important effect on running time. Therefore, compact-inverse schemes along the lines first proposed by Harry Markovitz of RAND have become increasingly important. Recently, two groups working independently, developed this approach with astounding results. For example, the Standard Oil Company of California group reports running-time on some of their typical large problems cut to 1/4.
>
> How to find the most compact inverse representation of a sparse matrix is still an unsolved problem:
>
> CONJECTURE: *If a non-singular matrix has K non-zero elements, it is always possible to represent them as a product of elementary matrices such that the total number of non-zero entries (excluding their diagonal unit elements) is at most K.* [*Incidentally, the empirical schemes just mentioned often have no more than K+10%K non-zeros in the inverse representation.*]

STATUS TO THE PRESENT (1980)

From 1967 onwards there has been an increasing interest in techniques for solving large-scale linear programs. A number of conferences have been exclusively concerned with the topic. Most general operations research and management science meetings have at least one session devoted to it. A selected reference list which I use in my seminars (mostly published during the period 1970-78) contain 237 titles which I have arranged by sub area.

General Books (10 exclusively large scale, 2 sparse methods, 8 other)	20
Survey articles	12
GUB, G-GUB and the decomposition principle	15
Variants of above	19
Block Triangularity	3
Linear optimal control and dynamic systems	14
Nested decomposition	4
Column generation, convex and nonlinear programs	34
Sparse matrix techniques	10
Large networks and related problems	37
Applications	52
Software	17
Total	237

Some idea of the recent research of the Systems Optimization Laboratory of the Operations Research Department at Stanford can be gleaned from the titles that follow:

Andre Perold: "Fundamentals of a Continuous Time Simplex Method".

Andre Perold and George B. Dantzig: "A Basis Factorization Method for Block Triangular Linear Programs".

Bob Fourer: "Solving Staircase-structured Linear Programs by Adaptation of the Simplex Method".

Ron Davis: "New Jump Conditions for State Constrained Optimal Control Problems".

Philip Abrahamson and George B. Dantzig: "Imbedded Dual Decomposition Approach to Staircase Systems".

John Birge: "Solving Staircase Systems under Uncertainty".

This Workshop may well mark the point in time when efficient methods for solving large dynamic systems may be more than just a promise. Thirty three years from the time the hope was first expressed that such methods be found, they may soon become a reality!

REFERENCES

[1] Wood, Marshall K., and Dantzig, George B., Programming of Interdependent Activities, I, *Econometrica,* July-October 1949.

[2] Dantzig, George B., Programming of Interdependent Activities, II, *Econometrica,* July-October 1949.

[3] Wood, M.K., and Geister, M.A., Development of Dynamic Models for Progr..m Planning, *Activity Analysis of Production and Allocation,* T. Koopmans, Ed., Wiley and Co. 1951.

[4] Dantzig, George B., Upper Bounds, Secondary Constraints and Block Triangularity in Linear Programming, *Econometrica,* 23, April 1955, pp 174-183.

[5] Dantzig, George B., Recent Advances in Linear Programming, *Management Science* Vol. 2, No. 2, January 1956, pp 131-144.

[6] Dantzig, George B., and Wolfe,Philip, Decomposition Principle for Linear Programs, *Operations Research* 8, Jan-Feb.1960, pp 101-111.
Another version of this paper appeared in *Econometrica* 29, Oct. 1960, pp 767-778.

[7] Dantzig, George B., Compact Basis Triangularization for the Simplex Method, *Recent Advances in Mathematical Programming,* Graves, R.L., and Wolfe, P., Eds., McGraw Hill, 1963, N.Y. pp 125-133.

[8] Dantzig, George B., Large-Scale Systems and the Computer Revolution, Proceedings of the Princeton Symposium on Mathematical Programming, H.W. Kuhn, Ed. Princeton University Press, Princeton N.J. Aug. 1967 pp 51-72.